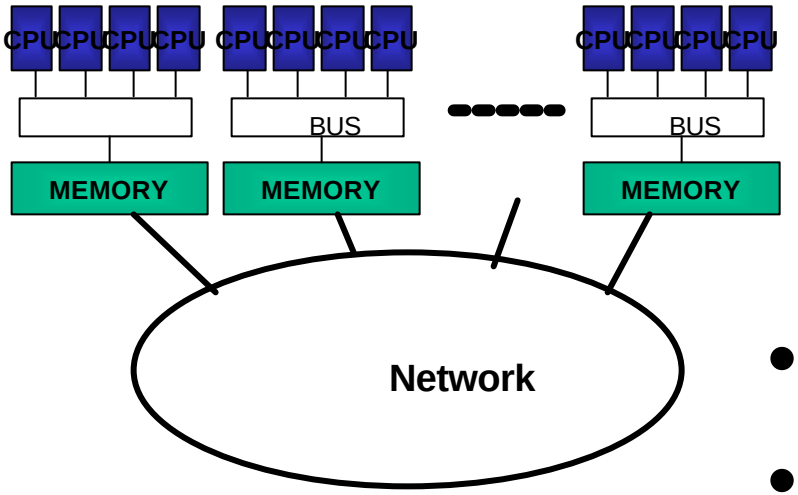
Porting MPI Applications to IBM SP

San Diego Supercomputing Center

Porting MPI Applications to IBM SP - Outline

- SP Architecture
  - CPU
  - Network
- Application data models compared to the T3e
- MPI issues
- Compilers
- Numerical and System Libraries
- Using SP Batch system
  - Covered in the talk “Running Jobs on the IBM SP”

General IBM SP Hardware



- **Power 3 processors**
- **Hybrid machine**
  - **N Shared Memory Processor Nodes**
  - **Each SMP node contains 2 to 8 processors sharing memory**
  - **Nodes connected together with a high speed network**

IBM SP network Architecture

- Bi-directional multi-stage interconnection network (omega-type)
  - Switch adaptor
  - Switch network
    - packet-switched
    - support for use by multiple processes
    - redundant routing paths
    - error detection
- Details of network architecture at: www.chips.ibm.com
- Differences between T3E
  - Message time (more) independent of placement
  - Nodes do not need to be contiguous
  - A little slower

NPACI IBM SP Configuration

- 144 IBM SP-High Nodes
  - 8 shared memory processors/node for 1152 processors
  - 6.4 Gbytes/second on-node memory bandwidth
  - 4 Gbytes/node memory
- Power3+ processors
  - 375 MHz
  - Up to 4 floating point operations per cycle
  - 1500 MFLOPS/processor (peak) for a total of ~1.73 Teraflops(peak)
  - 64KB L1 cache
  - 4 MB L2 cache
    - 2-way set associative
- Nodes connected together with IBM switch
  - 115 MB/second bi-directional
  - Maximum 4 MPI tasks/node when using switch (User Space)
  - Maximum 8 MPI tasks/node when using IP (for applications with low communication requirements)

NAVO IBM SP Configuration

- 334 IBM SP-High Nodes
  - 4 shared memory processors/node for total 1336 processors
  - 36 GB/node
  - 1.6 Gbytes/second into L2 cache at 100 MHz bus speed
- Power3 II processors
  - 375 MHz
  - Up to 4 floating point operations per cycle
  - 1.5 GFLOPS/processor for a total of 2.0 Teraflops (peak)
  - 64KB L1 cache
  - 4 MB L2 cache
    - Two-way set associative
- Nodes connected together with IBM switch
  - 300 MB/second peak, bi-directional

POWER3 Architecture

- Details of POWER3 architecture at:

www.chips.ibm.com/micronews/vol4_no4/powerpc.html

Cache structure T3E to SP

- Single-CPU optimizations - different cache architecture provides new optimization opportunities.
- Cray T3E
  - 8 KB Level 1 cache
  - 96 KB Level 2 cache
- IBM
  - 64KB Level 1 cache (128-way set associative)
  - 4 MB Level 2 cache (2-way set-associative)

Stick to standards to help portability

- Don’t rely on particular data sizes
  - Normal real data size may be 4 or 8 bytes
  - Double precision varies
  - Use ‘sequence” in Fortran derived types
  - MPI type sizes may change on different machines
- Watch out for large MPI tags
- Extensions to avoid
  - Cray Pointers
  - Non-Fortran 90 Namelist
  - Real*4 data types

Fortran Data Storage Models

- Default Fortran Data Types

Data Type	T3E Length (Bytes)	SP length (Bytes)
Character	1	1
Complex	2*8	2*4
Double Complex	2*8	2*8
Double Precision	8	8
Integer/Logical	8	4
Real	8	4

Cray T3E treats all real and integer variables as 8 byte quantities

For IBM you can force 8 and 8 byte reals
-qintsize=8 and -qrealsize=8 for similar data types
Reference - www.npaci.edu/BlueHorizon/porting.html

Moving MPI Applications to IBM SP

- Porting existing MPI Applications should be easy:
  - Recompile/relinked with IBM MPI-aware compilers/linkers
  - Replace other numerical library references with IBM equivalent
- Discuss reality:
  - Not too bad
  - Some data sizes are different
  - Different subset of MPI/IO
  - Buffer sizes are different
  - POE is used to run jobs

IBM MPI Features

- Standard MPI - IBM PSSP 2.4 Supports MPI 1.2 and parts of MPI-2 (mostly parallel I/O).
- 32-bit only - 64-bit version late in 2000?
- Best performance with message size > 1 Kbyte - aggregate smaller messages when possible
- Less "forgiving" than T3E implementation of bad or missing arguments
- Supports both User Space (US) network protocol and Internet protocol (IP) set with MP_EUILIB environment variable
- Can use shared-memory architecture for passing messages set with MPI_SHARED_MEMORY environment variable

Moving MPI Applications - Summary

- Re-compile source code with IBM “MPI-aware” compilers. Use POWER3 optimizations (-O3 -qarch=pwr3 -qtune=pwr3 -qstrict)
- Use IBM MASS library for common math functions - trig, log, exp, etc.
- Use IBM Numerical libraries - ESSL/PESSL where possible
- Check for program correctness
  - IBM may produce different results than other vendor’s machines
  - May have to modify source depending on data types used
- Tune program for IBM MPI
  - Message delivery implementation(s) - eager, rendezvous
  - Interdelay
  - Adjust buffer sizes

IBM MPI Environment Variables

- POE controls execution environment for MPI - important to have “correct” parameters and values for optimum performance
- There are many variables - check IBM poe manual for detailed information - however, some new ones not yet documented
  - MP_EUILIB=us [use high-speed network, User Space]
  - MP_SHARED_MEMORY=yes [have MPI use shared memory to streamline inter-task communication between processors within a node]
  - MP_INTRDELAY mystery parameter set to 100
  - MP_EAGER_LIMIT Changes the threshold value for message size, above which rendezvous protocol is used
  - MP_BUFFER_MEM Changes the maximum size of memory used by the communication subsystem to buffer early arrivals
  - MP_LABELIO Label IO with PID

Compilers

- mpXXX - “mp” denotes IBM’s MPI aware compiler shells
- mpcc [*options*] your_source.c (C)
  - “mpcc” is shell script - IBM MPI libraries linked automatically
- mpCC [*options*] your_source.C (C++)
- mpxlf [*options*] your_source.f (Fortran 77, only suffixes .f, .F)
- mpxlf90/95 [*options*] your_source.f (Fortran 90/95)
- Must use thread-safe compilers with MPI I/O - mpxlf90_r, mpcc_r, mpCC_r

Compiling

- Suggested Fortran compiler flags
  - “-O3” performs high-level optimizations
  - “-qstrict” used with -O3 to ensure compiler optimization does not alter program semantics
  - “-qarch=pwr3” produces an object that contains instructions that run on the POWER3 hardware platforms
  - “-qtune=pwr3” produces an object optimized for the POWER3 hardware platforms
  - “-bmaxdata:bytes” specifies max size reserved for program data segment “heap” (default is 128 MB)
  - “-bmaxstack:bytes” specifies max program stack size (default - 32 KB)

Compiling at NAVO

To compile codes using F90/F95 compilers on HABU, users must:

- Move their code to local (non-GPFS) /scratch filesystem on either of the two interactive nodes
- Or, if compiling with LoadLeveler in a batch run, move to the /scratch filesystem on one of the compute nodes.
- Once the compilation is complete - remove any files from the /scratch directory ASAP
- Only required for F90/F95 code and is due to lack of functionality in current version of GPFS.

Numerical Libraries

Use MASS for common mathematical functions - trig, exp, log, etc.

- MASS - mpxlf95 -O3 -qtune=pwr3 -qarch=pwr3 your_source.f -lmass

Use ESSL/PESSL for Linear Algebra, LAPACK, EISPACK, etc.

PESSL is MPI-based, runs in parallel. ESSL is single CPU and/or SMP-based.

- Common numerical functionality included - Linear Algebra, FFTs, Integration, Random number generation
- Most of LAPACK, ScaLAPACK included in ESSL/PESSL/BLAS mpxlf95 -O3 -qtune=pwr3 -qarch=pwr3 your_source.f -lblas -lessl -lpessl
- Convert SciLib calls to ESSL/PESSL equivalents

Porting Applications - References

- References
  - IBM RS/6000 SP References
    www.rs6000.ibm.com/resource/aix_resource/sp_books
  - Single CPU optimization information specific to the IBM architecture - “Scientific Applications in RS/6000 SP Environments" (www.redbooks.ibm.com/pubs/pdfs/redbooks/sg245611.pdf)
  - IBM RS/6000 Practical MPI Programming

NPACI IBM SP Blue Horizon Hardware Overview

- Machine name is horizon.npaci.edu
- Machine is
  - 144 nodes connected together with IBM High Speed switch
- Node is
  - 8 shared memory processors with a total of 4Gbytes/node
- Processor is
  - 375 MHz Power3
- Network is omega-type multi-stage
- A hybrid distributed/shared memory machine
  - Use MP (MPI or LAPI) for distributed memory nature
  - Use OpenMP/Pthreads for shared memory nature
  - Use Hybrid Method (MP + Threads) to exploit hybrid system architecture
- Disk
  - Local $HOME directory for each user - backed up
  - /work directory shared among users - not backed up
  - GPFS available, supports MPI-I/O
  - Total 5.1 TB

Available Fortran Data Storage Types

Data Type	Sub Type	T3E Length(Bytes)	SP Length(Bytes)
Complex	4	2 * 4	2 * 4
	8	2 * 8	2 * 8
	16	NA	2 * 16
Integer/Logical	1	1	4
	2	2	4
	4	4	4
	8	8	8
Real	4	4	4
	8	8	8
	16	NA	16